

EXHIBIT A

```

1  #import <appkit/Application.h>
2  #import <appkit/Window.h>
3  #import <appkit/Menu.h>
4  #import <appkit/MenuCell.h>
5  #import <appkit/Text.h>
6  #import <appkit/Form.h>
7  #import <soundkit/Sound.h>
8  #import <ldsyms.h>
9  #import <sys/loader.h>
10 #import <strings.h>
11 #import <libc.h>
12 #import "pathutil.h"
13 #import "a3DViewerView.h"
14 #import "CubeView.h"
15
16 static struct mach_header *header; // The header used for loading
17 static id customSubmenu; // Submenu handle for possible extension
18
19 static id inspectorPanel;
20 static id fileName;
21 static id parameterForm;
22 static id recordButton;
23 static id stopButton;
24 static id playButton;
25 static id deleteButton;
26 static id tmpSound;
27
28 @implementation a3DViewerView
29
30 + setCustomComponentData:(struct mach_header *)hd customMenu:(id)subMenu
31 {
32     id box;
33     id mCell;
34     id classObj;
35     header = hd;
36     customSubmenu = subMenu;
37     mCell = [customSubmenu addItem:@"3D Viewer"
38             action:@selector(showAlert)
39             keyEquivalent:0];
40     classObj = [[a3DViewerView alloc] init];
41     [mCell setTarget:classObj];
42     [customSubmenu display];
43
44     inspectorPanel = [NXApp loadNibSection:@"Inspector.nib" owner:classObj
45                     withNames:YES fromHeader:header];
46     box = NXGetNamedObject("3DFileBox", inspectorPanel);
47     fileName = NXGetNamedObject("3DFileName", box);
48     box = NXGetNamedObject("3DParameterBox", inspectorPanel);
49     parameterForm = NXGetNamedObject("3DForm", box);
50     [parameterForm setAction:@selector(setPhi) at:0];
51     [parameterForm setAction:@selector(setTheta) at:1];
52     [parameterForm setAction:@selector(setInvdist) at:2];
53

```

```

54     box = NXGetNamedObject("3DSoundButtonBox", inspectorPanel);
55     recordButton = NXGetNamedObject("3DRecordSoundButton", box);
56     [recordButton setAction:@selector(recordSound:)];
57     stopButton = NXGetNamedObject("3DStopSoundButton", box);
58     [stopButton setAction:@selector(stopSound:)];
59     playButton = NXGetNamedObject("3DPlaySoundButton", box);
60     [playButton setAction:@selector(playSound:)];
61     deleteButton = NXGetNamedObject("3DDeleteSoundButton", box);
62     [deleteButton setAction:@selector(deleteSound:)];
63     tmpSound = 0x0;
64     return classObj;
65 }
66
67 - (void) showAlert
68 {
69     NXRunAlertPanel(NULL, "Custom Menu Activated", NULL, NULL, NULL);
70 }
71
72 - initWithCustomComponent
73 {
74     self = [super init];
75     _3Dpanel = [NXApp loadNibSection:@"3DViewer.nib" owner:self
76                                     withNames:YES fromHeader:header];
77     _3DcontentView = [_3Dpanel contentView];
78
79     _3DcubeView = NXGetNamedObject("3DCubeView" , _3Dpanel);
80     [_3DcubeView setmachHeader:header];
81     [_3DcubeView setController:self];
82
83     [_3Ddatasets setTarget:self];
84     [_3Ddatasets setAction:@selector(openData:)];
85
86     return _3DcontentView;
87 }
88
89 - recordSound:sender
90 {
91     if( !tmpSound ) tmpSound = [Sound new];
92     [tmpSound setDelegate:self];
93     [tmpSound record];
94     return self;
95 }
96
97 - playSound:sender
98 {
99     if( tmpSound ) [tmpSound play];
100     else
101     [playButton setState:0];
102     return self;
103 }
104
105 - didPlay:sender
106 {

```

```

107     [playButton setState:0];
108     return self;
109 }
110
111 - didRecord:sender
112 {
113     [recordButton setState:0];
114     [_3DcubeView setViewSound:tmpSound];
115     return self;
116 }
117
118 - stopSound:sender
119 {
120     if( tmpSound ) [tmpSound stop];
121     return self;
122 }
123
124 - deleteSound:sender
125 {
126     [_3DcubeView deleteViewSound];
127     return self;
128 }
129
130 - inspectCustomComponent
131 {
132     [inspectorPanel makeKeyAndOrderFront:self];
133     [self refreshInspectorPanel];
134     return self;
135 }
136
137 - (void)refreshInspectorPanel
138 {
139     int index;
140     if( [inspectorPanel isVisible] ) {
141         [recordButton setTarget:self];
142         [stopButton setTarget:self];
143         [playButton setTarget:self];
144         [deleteButton setTarget:self];
145         [inspectorPanel orderFront:self];
146         if( [_3DcubeView currentDataFile] )
147             [fileName setStringValue:basename([_3DcubeView currentDataFile])];
148         [parameterForm setFloatValue:[_3DcubeView readPhi] at:0];
149         [parameterForm setFloatValue:[_3DcubeView readTheta] at:1];
150         [parameterForm setFloatValue:[_3DcubeView readInvdist] at:2];
151         for( index=0; index<3; index++ )
152             [parameterForm setTarget:self at:index];
153     }
154 }
155
156 - (void)setPhi
157 {
158     if( [_3DcubeView writePhi:[parameterForm floatValueAt:0]] )
159         [parameterForm setFloatValue:[_3DcubeView readPhi] at:0];;

```

```

160     [parameterForm selectTextAt:0];
161 }
162
163 - (void)setTheta
164 {
165     if( [_3DcubeView writeTheta:[parameterForm floatValueAt:1]] )
166         [parameterForm setFloatValue:[_3DcubeView readTheta] at:1];;
167     [parameterForm selectTextAt:1];
168 }
169
170 - (void)setInvdist
171 {
172     if( [_3DcubeView writeInvdist:[parameterForm floatValueAt:2]] )
173         [parameterForm setFloatValue:[_3DcubeView readInvdist] at:2];;
174     [parameterForm selectTextAt:2];
175 }
176
177 - openData:sender
178 {
179     [_3DcubeView openData:sender];
180     [self perform:@selector(displayAndFlush:) with:self
181              afterDelay:1 cancelPrevious:YES];
182     return self;
183 }
184
185 - (void) displayAndFlush:sender
186 {
187     [[_3DcontentView window] disableFlushWindow];
188     [[_3DcontentView window] display];
189     [[_3DcontentView window] reenableViewFlushWindow];
190     [[_3DcontentView window] flushWindowIfNeeded];
191     [self resetFirstResponder:_3DcubeView];
192 }
193
194 - setViews:(id)cv
195 {
196     _3DcubeView = cv;
197     _3DcontentView = [cv superview];
198     return self;
199 }
200
201 - (void)resetFirstResponder:(id)frv
202 {
203     id vt = [_3DcontentView superview];
204
205     if ( [vt isKindOfClass:[Text class]] )
206         [vt textDidGetKeys:vt isEmpty:NO];
207     [[frv window] makeFirstResponder:frv];
208     [self refreshInspectorPanel];
209 }
210
211 - free
212 {

```

```
213         if( tmpSound ) [tmpSound free];
214         return [super free];
215     }
216
217     - write:(NXTypedStream *)stream
218     {
219         [super write:stream];
220         return self;
221     }
222
223     - read:(NXTypedStream *)stream
224     {
225         [super read:stream];
226         return self;
227     }
228
229     - awake
230     {
231         [super awake];
232         return self;
233     }
234
235     @end
236
```